AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings of claims in the application:

LISTING OF CLAIMS

1. (Currently Amended) A small footprint device comprising:

at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object-oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized; and

a global data structure for permitting one program module to access information from another program module by bypassing said context barrier.

2-22. (Cancelled)

23. (Previously Presented) The small footprint device of claim 1 in which said context barrier allocates separate name spaces for each program module.

24. (Previously Presented) The small footprint device of claim 23 in which at least two program modules can access said global data structure even though they are located in different respective name spaces.

25. (Previously Presented) The small footprint device of claim 1 in which said context barrier allocates separate memory spaces for each program module.

26. (Previously Presented) The small footprint device of claim 25 in which at least two program modules can access said global data structure even though they are located in different respective memory spaces.

27. (Previously Presented) The small footprint device of claim 1 wherein said program modules comprise at least one of a principal or an object and wherein said context barrier enforces security checks on at least one of a principal, an object and an action.

28. (Previously Presented) The small footprint device of claim 27 in which at least one security check is based on partial name agreement between a principal and an object.

29. (Previously Presented) The small footprint device of claim 28 in which at least one program can access said global data structure without said at least one security check.

30. (Previously Presented) The small footprint device of claim 27 in which at least one security check is based on memory space agreement between a principal and an object.

31. (Previously Presented) The small footprint device of claim 30 in which at least one program can access a global data structure without said at least one security check.

32. (Currently Amended) A method of operating a small footprint device that includes a processing machine, wherein program modules are executed on the processing machine, the method comprising:

separating contexts using a context barrier, said context barrier configured to ~~use said memory to control object-oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said separating further comprising:

preventing said access if said access is unauthorized; and

enabling said access if said access is authorized;

executing groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~; and

permitting access to information across said context barrier by bypassing said context barrier using a global data structure.

33. (Previously Presented) The method of claim 32 wherein said program modules comprise at least one of a principal or an object and wherein the context barrier will not permit a principal to perform an action on an object unless both principal and object are part of the same context unless the request is for access to a global data structure.

34. (Currently Amended) A method of permitting access to information on a small footprint device from a first program module to a second program module separated by a context barrier, said small footprint device comprising:

[illegible]

at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized, the method comprising:

creating a global data structure which may be accessed by at least two program modules; and

using said global data structure to permit access to information across said context barrier by bypassing said context barrier.

[illegible]

35. (Currently Amended) A method of communicating across a context barrier separating program modules on a small footprint device, said small footprint device comprising:

at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized, the method comprising:

creating a global data structure;

permitting at least one program module to write information to said global data structure; and

having at least one other program module read information from said global data structure, bypassing said context barrier.

36. (Currently Amended) A computer program product, comprising:

a memory medium; and

a computer controlling element comprising instructions for implementing a context barrier on a small footprint device and for bypassing said context barrier using a global data structure, said small footprint device comprising:

at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

37. (Previously Presented) The computer program product of claim 36 in which said medium is a carrier wave.

38. (Currently Amended) A computer program product, comprising:

a memory medium; and

a computer controlling element comprising instructions for separating a plurality of programs on a small footprint device by running them in respective contexts and for permitting one program to access information from another program by bypassing a context barrier using a global data structure, said small footprint device comprising:

[illegible]

at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object-oriented access of a program module executing in one context to information and/or a program module~~

~~executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

39. (Previously Presented) The computer program product of claim 38 in which said medium is a carrier wave.

40. (Currently Amended) A carrier wave carrying instructions for implementing a global data structure for bypassing a context barrier on a small footprint device over a communications link, said small footprint device comprising:

at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object-oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

41. (Currently Amended) A carrier wave carrying instructions over a communications link for separating a plurality of programs on a small footprint device by running them in respective contexts and for permitting one program to access information from another program using at least one global data structure, said small footprint device comprising:

at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object-oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

42. (Currently Amended) A method of transmitting code over a network, comprising transmitting a block of code from a server, said block of code comprising instructions for implementing a global data structure for bypassing a context barrier on a small footprint device over a communications link, said small footprint device comprising:

at least one processing element configured to execute groups of one or more program modules in separate contexts, said one or more program modules comprising zero or more sets of executable instructions and zero or more sets of data definitions, said zero or more sets of executable instructions and said zero or more data definitions grouped as object definitions, each context comprising a protected object instance space such that at least one of said object definitions is instantiated in association with a particular context ~~objects of a program module associated with a particular context~~;

a memory comprising instances of objects; and

a context barrier for separating and isolating said contexts, said context barrier configured ~~to use said memory to control object-oriented access of a program module executing in one context to information and/or a program module executing in another context~~ for controlling execution of at least one instruction of one of said zero or more sets of instructions comprised by a program module based at least in part on whether said at least one instruction is executed for an object instance associated with a first one of said one or more separate contexts and whether said at least one instruction is requesting access to an instance of an object definition associated with a second one of said one or more separate contexts, said context barrier further configured to prevent said access if said access is unauthorized and enable said access if said access is authorized.

43. (New) The small footprint device of claim 1 wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

44. (New) The small footprint device of claim 1 wherein

said memory comprises object header data, said object header data comprising information associated with at least one of said instances of objects; and

said controlling execution is based at least in part on said object header data.

45. (New) The small footprint device of claim 1 wherein

said memory is partitioned into a plurality of memory spaces with instances of objects allocated for storage in one of said plurality of storage spaces; and

said controlling execution is based at least in part on determining the storage space allocated to an executing object instance and an accessed object instance.

46. (New) The method of claim 32 wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

47. (New) The method of claim 32 wherein said controlling execution is based at least in part on object header data comprising information associated with at least one of said instances of objects.

48. (New) The method of claim 32 wherein

a memory of said small footprint device is partitioned into a plurality of memory spaces with instances of objects allocated for storage in one of said plurality of storage spaces; and

said controlling execution is based at least in part on determining the storage space allocated to an executing object instance and an accessed object instance.

49. (New) The method of claim 34 wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

50. (New) The method of claim 34 wherein said controlling execution is based at least in part on object header data comprising information associated with at least one of said instances of objects.

51. (New) The method of claim 34 wherein

a memory of said small footprint device is partitioned into a plurality of memory spaces with instances of objects allocated for storage in one of said plurality of storage spaces; and

said controlling execution is based at least in part on determining the storage space allocated to an executing object instance and an accessed object instance.

52. (New) The method of claim 35 wherein an object instance is associated with a context by recording the name of said context in a header of said object instance, information in said header inaccessible to said one or more program modules.

53. (New) The method of claim 35 wherein said controlling execution is based at least in part on object header data comprising information associated with at least one of said instances of objects.

54. (New) The method of claim 35 wherein

a memory of said small footprint device is partitioned into a plurality of memory spaces with instances of objects allocated for storage in one of said plurality of storage spaces; and

said controlling execution is based at least in part on determining the storage space allocated to an executing object instance and an accessed object instance.